

Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/US04/040442

International filing date: 03 December 2004 (03.12.2004)

Document type: Certified copy of priority document

Document details: Country/Office: US
Number: 60/527,900
Filing date: 05 December 2003 (05.12.2003)

Date of receipt at the International Bureau: 17 January 2005 (17.01.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)



World Intellectual Property Organization (WIPO) - Geneva, Switzerland
Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse



THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

January 08, 2005

THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM THE RECORDS OF THE UNITED STATES PATENT AND TRADEMARK OFFICE OF THOSE PAPERS OF THE BELOW IDENTIFIED PATENT APPLICATION THAT MET THE REQUIREMENTS TO BE GRANTED A FILING DATE.

APPLICATION NUMBER: 60/527,900

FILING DATE: *December 05, 2003*

RELATED PCT APPLICATION NUMBER: *PCT/US04/40442*



Certified By

Jon W Dudas

Under Secretary
of Commerce for Intellectual Property
and Acting Director of the
United States Patent and Trademark Office

18351 U.S. PTO
120503

PTO/SB/16 (08-03)

Approved for use through 07/31/2006. OMB 0651-0032


U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PROVISIONAL APPLICATION FOR PATENT COVER SHEET

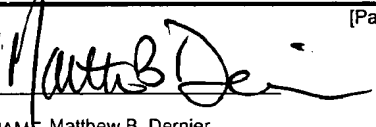
This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53(c).

Express Mail Label No. **EL 984296134US**

| INVENTOR(S) | | | | | |
|--|--|--|--|---|--|
| Given Name (first and middle (if any)) | | Family Name or Surname | | Residence (City and either State or Foreign Country) | |
| Yun-Qing Dekun Zou | | Shi Zou | | Millburn, NJ Kearny, NJ | |
| Additional inventors are being named on the _____ 1 _____ separately numbered sheets attached hereto | | | | | |
| TITLE OF THE INVENTION (500 characters max) | | | | | |
| ROBUST LOSSLESS IMAGE DATA HIDING IN INTEGER WAVELET DOMAIN | | | | | |
| Direct all correspondence to: CORRESPONDENCE ADDRESS | | | | | |
| <input checked="" type="checkbox"/> Customer Number:  27538 | | | | | |
| OR | | | | | |
| <input type="checkbox"/> Firm or Individual Name | | PATENT TRADEMARK OFFICE | | | |
| Address | | | | | |
| Address | | | | | |
| City | | State | | Zip | |
| Country | | Telephone | | Fax | |
| ENCLOSED APPLICATION PARTS (check all that apply) | | | | | |
| <input checked="" type="checkbox"/> Specification Number of Pages 32 | | <input type="checkbox"/> CD(s), Number _____ | | | |
| <input type="checkbox"/> Drawing(s) Number of Sheets _____ | | <input type="checkbox"/> Other (specify) _____ | | | |
| <input type="checkbox"/> Application Date Sheet. See 37 CFR 1.76 | | | | | |
| METHOD OF PAYMENT OF FILING FEES FOR THIS PROVISIONAL APPLICATION FOR PATENT | | | | | |
| <input checked="" type="checkbox"/> Applicant claims small entity status. See 37 CFR 1.27. | | FILING FEE Amount (\$) | | | |
| <input checked="" type="checkbox"/> A check or money order is enclosed to cover the filing fees. | | <div>\$80.00</div> | | | |
| <input checked="" type="checkbox"/> The Director is hereby authorized to charge filing fees or credit any overpayment to Deposit Account Number: 11-0223 | | | | | |
| <input type="checkbox"/> Payment by credit card. Form PTO-2038 is attached. | | | | | |
| The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government. | | | | | |
| <input checked="" type="checkbox"/> No. | | | | | |
| <input type="checkbox"/> Yes, the name of the U.S. Government agency and the Government contract number are: _____ | | | | | |

17548 U.S. PTO
60/527900

120503

Respectfully submitted,  [Page 1 of 2] Date 12/5/03

SIGNATURE _____ REGISTRATION NO. 40,989
(if appropriate)
TYPED or PRINTED NAME Matthew B. Dernier Docket Number: 436/10

TELEPHONE (732) 634-7634

USE ONLY FOR FILING A PROVISIONAL APPLICATION FOR PATENT

This collection of information is required by 37 CFR 1.51. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 8 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Mail Stop Provisional Application, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Additional Pag

PTO/SB/16 (08-03)

Approved for use through 07/31/2006. OMB 0651-0032

U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

Docket Number 436/10

| INVENTOR(S)/APPLICANT(S) | | |
|---|-------------------|---|
| Given Name (first and middle [if any]) | Family or Surname | Residence (City and either State or Foreign Country) |
| Zhicheng | Ni | Karny, NJ |

[Page 2 of 2]

Number _____ of _____

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Problem Image Mailbox.**

TITLE OF THE INVENTION

Robust Lossless Image Data Hiding in Integer Wavelet Domain

Background

This invention relates to image data hiding in integer wavelet domain that can recover the original images without any distortion after the hidden data have been retrieved from the stego-images, and that is robust against JPEG2000 compression. In addition, it does not generate salt-pepper noise at all.

Data hiding is referred to as a process to embed useful data (information) into a cover media. It has wide applications for the purpose of identification, annotation, copyright protection and authentication. In these applications, people do care about the cover media. That is, the hidden data and the cover media are closely related. This type of data hiding is often referred to as watermarking.

For this type of data hiding, the hidden data are required to be perceptually transparent. In other words, we require the marked media to be as similar to the cover media as possible.

In most of cases, the cover media will experience some distortion and cannot be inverted back to the original media data. That is, some permanent distortion exists even after the hidden data have been extracted. For instance, round-off error, truncation error and quantization error make lossless data hiding impossible. However, for some applications such as medical diagnosis and law enforcement, it is desired to invert the marked media back to the original cover media after the hidden data have been retrieved. The marking techniques satisfying this requirement are referred to as *lossless*, sometimes as *distortionless*. They are also called *reversible* marking techniques. This type of marking techniques is suitable for applications where the exact original media data should be recovered.

Recently, some lossless marking techniques have been reported in the literature. The first method [honsinger 99] is carried out in the image spatial domain. Another spatial domain technique was reported in [fridrich 01]. There also exists a lossless marking technique in the transform domain [macq 99]. From our study of the transform domain method, the upper bound of the amount of hidden data is estimated to be 2000 bits (equivalent to 250 bytes) for a $512 \times 512 \times 8$ image. Hence, these techniques are not suitable for applications where a much larger amount of data is requested to hide in images. The capacity of the method reported in [vleeschouwer 01] is also very limited except it exhibits robustness against high quality JPEG compression. These techniques aim at authentication, instead of data hiding. As a result, the amount of hidden data is quite limited.

The first lossless marking technique that is suitable for high embedding rate data hiding was presented in [goljan 01]. Its main idea is as follows. The pixels in an image

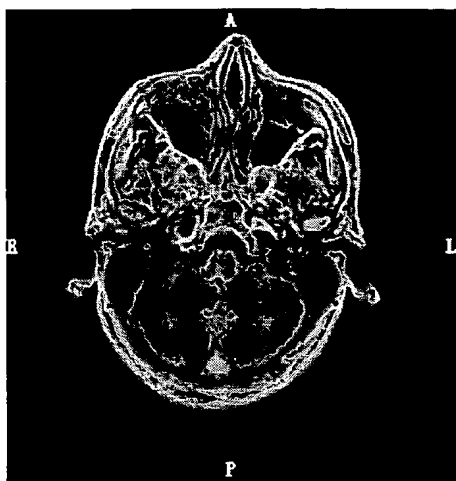
are divided into non-overlapped blocks, each consisting of a number of adjacent pixels. For instance, it could be a horizontal block having four consecutive pixels. A discrimination function is established to classify the blocks into three different categories, Regular, Singular and Unusable. (The authors used the discrimination function to capture the smoothness of the groups.) An invertible operation can be applied to groups. That is, it can map a gray level value to another gray level value. It is reversible since applying it to a gray level value twice produces the original gray level value. This invertible operation is hence called *flipping*. For typical images, flipping with a small amplitude will lead to an increase of the discrimination function, resulting in more Regular groups and less Singular groups. It is this bias that enables lossless data hiding. While it is novel, and successful in lossless data hiding, the amount of hidden data by this technique is still not large enough for certain applications. The pay-load was estimated to be in a range from 3,000 bits to 24,000 bits for a $512 \times 512 \times 8$ gray image according to [goljan 01]. Another problem with the method is that when the capacity increases, the visual quality will drop severely. For instance, PSNR drops to as low as 35 dB. Sometimes, unpleasant artifacts may occur.

The method based on the integer wavelet transform [xuan 02] is a recently proposed reversible data hiding technique that can achieve a quite large capacity. Its main idea is as follows. After the integer wavelet transform is applied to the original image, the *bias* between binary 1s and 0s in the bit-planes of the subbands LH, HL, HH is largely increased. Hence, the 1s and 0s in these bit-planes can be losslessly compressed to leave a large space for data hiding. After data embedding, inverse integer wavelet transform is applied to form the marked image. The capacity achieved in this technique is quite large. The PSNR of the marked image is, however, not high due to the histogram modification applied in order to avoid overflow/underflow. For instance, the PSNR is only 28 dB for a few images.

The method based on histogram manipulation [ni 03] is a newly invented lossless data hiding technique, which can embed a large amount of data (5k-80k bits for a $512 \times 512 \times 8$ grayscale image) while keeping the high visual quality (the PSNR is guaranteed to be above 48 dB) for a vast majority of images.

Among all of these lossless data hiding techniques, however, there is only one existing lossless data hiding technique that can be robust against compression applied to the stego-image [vleeschouwer 01]. That is, only with the technique discussed in [vleeschouwer 01] the hidden data can still be extracted out correctly after the stego-media have gone through compression within a reasonable extent. For all of the rest techniques, the hidden data can not be recovered after stego-media compression.

When it is robust to compression, it generates annoying salt-pepper noise because it uses module 256 addition. That is, when the pixel gray value is close to 256 (brightest) and/or 0 (darkest), the modulo 256 addition will likely cause flipping over between the brightest and darkest gray values. This often happens with medical images. One example is shown in Figure 1, where Figure 1 (a) is an original medical image, and Figure 1 (b) a stego-image. Another example is shown in Figure 2. Obviously, this type of salt-pepper noise is annoying and not acceptable for many applications



(a)

Figure 1. (a) Original medical image



(b)

(b) stego-image with salt-pepper noise.



Figure 2. (a) Original N1A image



(b) Stego-image with salt-pepper noise.

Summary Of The Invention

The invented method and apparatus provide a novel and more advanced lossless data hiding technique that 1) avoids salt-pepper noise; 2) is more robust against JPEG2000 compression than the only existing lossless data hiding that can resist compression [vleeschouwer 01]; 3) can be integrated into JPEG2000 standard by working on integer wavelet domain.

Detailed Description Of The Invention

The method is based on the following statistics. In integer wavelet domain, HL or LH sub-band is divided into non-overlapping blocks. Study shows that the mean value of all the coefficients in any block tends to be zero or close to zero. By manipulating the mean values, information can be embedded into the image. To avoid overflow and underflow, we manipulate LL sub-band to offset the influence of the change to HL or LH sub-band. We invented our lossless data hiding technique that does not generate salt-pepper noise and is robust against JPEG2000 compression. Channel coding technique such as the BCH codes is applied to enhance robustness. It also uses shuffling technique.

In this report, we propose a new lossless data hiding framework in integer wavelet domain. The original cover image can be recovered exactly after the hidden data are extracted out and the hidden data are robust to JPEG2000 compression.

We find that the HL and LH sub-bands coefficients statistically have characteristics of zero-mean and Laplacian-like distribution. If we divide a sub-band, for example HL1, into non-overlapping blocks, the mean values of coefficients of these blocks will have zero-mean and Laplace-like distribution. Figure 1 illustrates mean value distribution of N1A's HL1 sub-band in red color plane, where X-axis is mean value. Y-axis is number of blocks that have a particular mean value.

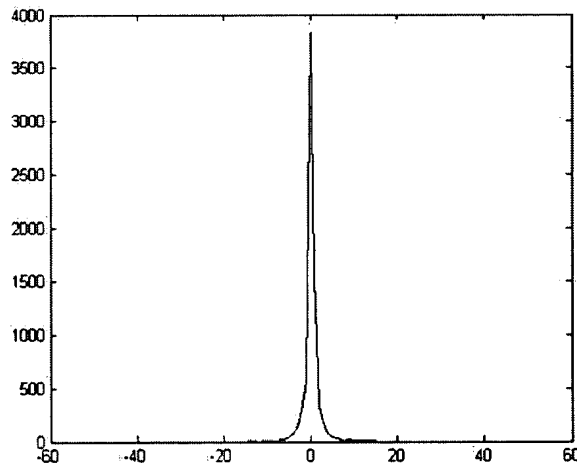


Figure 1: Mean values distribution of 10x10 blocks in HL1 sub-band of N1A (red color plane)

Our watermarking method is to first divide HL1 or LH1 sub-band into non-overlapping blocks of N by N . Then, to embed '1', we simply shift the mean value of the coefficients in a block by an amount, S . As a result, the mean value of this block will shift by S . To embed '0', we keep the coefficients in a block unchanged.

In lossless data hiding, one crucial problem is that after manipulations are made to original image, the pixel values of marked image may exceed the permitted range. This is

because the pixel value is represented by a fixed number of bits. (Usually, 8 bits is used to represent a color plane.) In this case, the exceeded part will be truncated. As a result, the image can no longer be recovered to the original one. The reversibility will be lost. This is called overflow and underflow problem.

We study the effect of the manipulation of sub-bands coefficients on the pixel values of an image and find that we can prevent overflow and/or underflow by manipulating the corresponding coefficients in LL sub-band. Below is the theoretical proof of our proposed approach.

In JPEG2000 standard, 5/3 integer wavelet filter is used as the default filter for reversible image encoding [1][2]. 5/3 filter is first proposed by Le Gall et. in [3]. The coefficients of the 5/3 wavelet filter are given in Table 1 [1].

Table 1: 5/3 Filter Coefficients

| i | Analysis Filter Coefficients | | Synthesis Filter Coefficients | |
|-----------|------------------------------|------------------------------|-------------------------------|------------------------------|
| | Low-pass Filter $h_L(i)$ | High-pass Filter $h_H(i)$ | Low-pass Filter $g_L(i)$ | High-pass Filter $g_H(i)$ |
| 0 | 6/8 | 1 | 1 | 6/8 |
| +1 and -1 | 2/8 | -1/2 | 1/2 | -2/8 |
| +2 and -2 | -1/8 | | | -1/8 |

We want to embed data into a sub-band of the integer wavelet transform (IWT) of an image. In this case, the effect of changing the wavelet coefficients on the spatial domain counterpart needs to be investigated.

In the Z transform, the synthesis filter can be represented as follows.

Low-pass filter:

$$G_L(z) = \frac{1}{2}z^{-1} + 1 + \frac{1}{2}z$$

High-pass filter:

$$G_H(z) = -\frac{1}{8}z^{-2} - \frac{2}{8}z^{-1} + \frac{6}{8} - \frac{2}{8}z - \frac{1}{8}z^2$$

For 2-D image, we have,

Low-pass filter in x axis:

$$G_{Lx}(z_x) = \frac{1}{2}z_x^{-1} + 1 + \frac{1}{2}z_x$$

High-pass filter in x axis:

$$G_{Hx}(z_x) = -\frac{1}{8}z_x^{-2} - \frac{2}{8}z_x^{-1} + \frac{6}{8} - \frac{2}{8}z_x - \frac{1}{8}z_x^2$$

Low-pass filter in y axis:

$$G_{Ly}(z_y) = \frac{1}{2}z_y^{-1} + 1 + \frac{1}{2}z_y$$

High-pass filter in y axis:

$$G_{Hy}(z_y) = -\frac{1}{8}z_y^{-2} - \frac{2}{8}z_y^{-1} + \frac{6}{8} - \frac{2}{8}z_y - \frac{1}{8}z_y^2$$

For HL sub-band, the synthesis filter is to first apply the high-pass to the row (x direction), then apply low-pass to the column (y direction). The result is:

$$\begin{aligned}
G_{HL}(z_x, z_y) &= G_{Hx}(z_x) \times G_{Ly}(z_y) \\
&= \left(-\frac{1}{8}z_x^{-2} - \frac{2}{8}z_x^{-1} + \frac{6}{8} - \frac{2}{8}z_x - \frac{1}{8}z_x^2\right) \times \left(\frac{1}{2}z_y^{-1} + 1 + \frac{1}{2}z_y\right) \\
&= -\frac{1}{16}z_x^{-2}z_y^{-1} - \frac{1}{8}z_x^{-2} - \frac{1}{16}z_x^{-2}z_y - \frac{1}{8}z_x^{-1}z_y^{-1} - \frac{2}{8}z_x^{-1} - \frac{1}{8}z_x^{-1}z_y \\
&\quad + \frac{3}{8}z_y^{-1} + \frac{6}{8} + \frac{3}{8}z_y - \frac{1}{8}z_xz_y^{-1} - \frac{2}{8}z_x - \frac{1}{8}z_xz_y - \frac{1}{16}z_x^2z_y^{-1} - \frac{1}{8}z_x^2 - \frac{1}{16}z_x^2z_y
\end{aligned}$$

For LH sub-band, the synthesis filter is to first apply the high-pass to the columns (y direction), then apply low-pass to the rows (x direction). The result is:

$$\begin{aligned}
G_{LH}(z_x, z_y) &= G_{Lx}(z_x) \times G_{Hy}(z_y) \\
&= \left(-\frac{1}{8}z_y^{-2} - \frac{2}{8}z_y^{-1} + \frac{6}{8} - \frac{2}{8}z_y - \frac{1}{8}z_y^2\right) \times \left(\frac{1}{2}z_x^{-1} + 1 + \frac{1}{2}z_x\right) \\
&= -\frac{1}{16}z_y^{-2}z_x^{-1} - \frac{1}{8}z_y^{-2} - \frac{1}{16}z_y^{-2}z_x - \frac{1}{8}z_y^{-1}z_x^{-1} - \frac{2}{8}z_y^{-1} - \frac{1}{8}z_y^{-1}z_x \\
&\quad + \frac{3}{8}z_x^{-1} + \frac{6}{8} + \frac{3}{8}z_x - \frac{1}{8}z_yz_x^{-1} - \frac{2}{8}z_y - \frac{1}{8}z_yz_x - \frac{1}{16}z_y^2z_x^{-1} - \frac{1}{8}z_y^2 - \frac{1}{16}z_y^2z_x
\end{aligned}$$

For LL sub-band, apply low-pass filter to both directions. The result is:

$$\begin{aligned}
G_{LL}(z_x, z_y) &= G_{Lx}(z_x) \times G_{Ly}(z_y) \\
&= \left(\frac{1}{2}z_x^{-1} + 1 + \frac{1}{2}z_x\right) \times \left(\frac{1}{2}z_y^{-1} + 1 + \frac{1}{2}z_y\right) \\
&= \frac{1}{4}z_x^{-1}z_y^{-1} + \frac{1}{2}z_x^{-1} + \frac{1}{4}z_x^{-1}z_y + \frac{1}{2}z_y^{-1} + 1 + \frac{1}{2}z_y + \frac{1}{4}z_xz_y^{-1} + \frac{1}{2}z_x + \frac{1}{4}z_xz_y
\end{aligned}$$

It is much easier to understand if we rewrite the filter coefficients in the following matrix form:

$$G_{HL} = \begin{bmatrix} -\frac{1}{16} & -\frac{1}{8} & \frac{3}{8} & -\frac{1}{8} & -\frac{1}{16} \\ -\frac{1}{8} & -\frac{2}{8} & \frac{6}{8} & -\frac{2}{8} & -\frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{3}{8} & \frac{1}{8} & \frac{1}{16} \end{bmatrix}$$

$$G_{LH} = \begin{bmatrix} -\frac{1}{16} & -\frac{1}{8} & -\frac{1}{16} \\ -\frac{1}{8} & -\frac{2}{8} & -\frac{1}{8} \\ \frac{3}{8} & \frac{6}{8} & \frac{3}{8} \\ -\frac{1}{8} & -\frac{2}{8} & -\frac{1}{8} \\ -\frac{1}{16} & -\frac{1}{8} & -\frac{1}{16} \end{bmatrix} \quad G_{LL} = \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix}$$

According to [4], lifting scheme is a substitute of wavelet transform because of its advantage in efficient implementation. The impulse response with respect to an HL coefficient at (i,j) is centered at $(2i-1, 2j)$ position in the spatial domain. For an LH coefficient at (i, j) , the center is at $(2i, 2j-1)$ in the spatial domain. For an LL coefficient at (i, j) , the center is at $(2i-1, 2j-1)$ in the spatial domain.

First, let us consider a unit input into an HL1 coefficient with position (i,j) , denoted by $\overline{C_{HL}(i,j)}$. Denote the output response in spatial domain by $k(\overline{i,j})$, where k represents the amplitude, $\overline{(i,j)}$ indicates the coordinates of the point affected in the spatial domain. The unit input described above and its response in the spatial domain are represented in the next equation.

$$\begin{aligned} \overline{C_{HL}(i,j)} \Leftrightarrow & -\frac{1}{16}\overline{(2i-2, 2j-2)} - \frac{1}{8}\overline{(2i-1, 2j-2)} - \frac{1}{16}\overline{(2i, 2j-2)} - \frac{1}{8}\overline{(2i-2, 2j-1)} \\ & - \frac{2}{8}\overline{(2i-1, 2j-1)} - \frac{1}{8}\overline{(2i, 2j-1)} + \frac{3}{8}\overline{(2i-2, 2j)} + \frac{6}{8}\overline{(2i-1, 2j)} \\ & + \frac{3}{8}\overline{(2i, 2j)} - \frac{1}{8}\overline{(2i-2, 2j+1)} - \frac{2}{8}\overline{(2i-1, 2j+1)} - \frac{1}{8}\overline{(2i, 2j+1)} \\ & - \frac{1}{16}\overline{(2i-2, 2j+2)} - \frac{1}{8}\overline{(2i-1, 2j+2)} - \frac{1}{16}\overline{(2i, 2j+2)} \end{aligned}$$

In the above equation, $-\frac{1}{16}\overline{(2i-2, 2j-2)}$ means that $-\frac{1}{16}$ is the response caused by the unit input to the HL1 coefficient at (i, j) , $\overline{(2i-2, 2j-2)}$ means the response is located at $(2i-2, 2j-2)$ in the spatial domain.

If a unit input is applied to an LL coefficient positioned at (i, j) , the corresponding response is:

$$\overline{C_{LL}(i,j)} \Leftrightarrow$$

$$\begin{aligned}
& \frac{1}{4}\overline{(2i-2, 2j-2)} + \frac{1}{2}\overline{(2i-2, 2j-1)} + \frac{1}{4}\overline{(2i-2, 2j)} \\
& + \frac{1}{2}\overline{(2i-1, 2j-2)} + \overline{(2i-1, 2j-1)} + \frac{1}{2}\overline{(2i-1, 2j)} \\
& + \frac{1}{4}\overline{(2i, 2j-2)} + \frac{1}{2}\overline{(2i, 2j-1)} + \frac{1}{4}\overline{(2i, 2j)}
\end{aligned}$$

We can verify that:

$$\overline{C_{HL}(i, j)} + \frac{1}{4}\overline{C_{LL}(i, j)} + \frac{1}{4}\overline{C_{LL}(i, j+1)} \Leftrightarrow \frac{1}{2}\overline{(2i-2, 2j)} + \overline{(2i-1, 2j)} + \frac{1}{2}\overline{(2i, 2j)}$$

It can be seen that if we change an HL1 coefficient at (i,j) by S and change both LL1 coefficients at (i,j) and (i,j+1) by S/4, the pixel value change in the spatial domain will have the same sign as S.

Similarly, for a coefficient at LH sub-band, we can get

$$\overline{C_{LH}(i, j)} + \frac{1}{4}\overline{C_{LL}(i, j)} + \frac{1}{4}\overline{C_{LL}(i+1, j)} \Leftrightarrow \frac{1}{2}\overline{(2i, 2j-2)} + \overline{(2i, 2j-1)} + \frac{1}{2}\overline{(2i, 2j)}$$

Embedding

Data can be embedded into sub-band HL1 or LH1. We will explain our method assuming HL1 is used. At first, HL1 sub-band is divided into non-overlapped blocks of N by N. In each block, the mean value is either 0 or close to zero. Equivalently, the absolute value of the mean should be less than a threshold 'T'. The basic idea is if a bit '1' is going to be embedded, the absolute mean value of the block will be shifted away from 0 by a quantity larger than 'T'. Here, we do not change every coefficients of this block.

Instead, we use the following mask

$$\begin{pmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 \\
0 & 1 & 1 & 1 & 0 \\
0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0
\end{pmatrix}$$

where block size is N=5 for illustration purpose. We only change those coefficients whose positions are associated with '1' in the above mask. In other words, we keep the outer bound of the block unchanged. By doing this, the effect caused by changing coefficients in one block will not interfere with that caused by changing coefficients in adjacent blocks. If we modify a coefficient at (i, j) in HL sub-band by a quantity S (S can be positive or negative), we will modify the corresponding two coefficients of LL sub-bands at (i, j) and (i, j+1) both by a quantity S/4. According to the previous proof, all the affected pixels in the spatial domain will either increase(if S>0) or decrease(if S<0).

When embedding a bit into a block in HL or LH sub-band, we first find out the corresponding spatial block that could possibly be affected by embedding the bit. Then the pixel values of the block is checked and classified into four categories.

A: have no pixel values near 255 and 0

B: have pixel values near 255 but have no pixel values near 0

C: have pixel values near 0 but have no pixel values near 255

D: have pixel values near 0 and have pixel values near 255

If '0' is to be embedded into a block, we simply keep the block unchanged regardless of which category this block falls into.

If '1' is to be embedded, we apply different embedding rules to different categories.

For A:

In this case, overflow and underflow will not take place. We can either increase the mean value or decrease the mean value of the coefficients in the block.

For B:

We only decrease the mean by S. As is shown before, the pixel values in spatial domain will only decrease after changes are made. Thus, no pixel value will exceed the border that is 255.

For C:

We only increase the mean by S. As is shown before, the pixel values will only increase after changes are made. Thus, no pixel value will exceed the border that is 0.

For D:

We do not change the coefficients in this case. It is equal to a bit '0' is embedded here. We rely on ECC coding to rectify it.

From the above analysis, overflow and underflow problems are overcome.

Extraction and original image recovering

Hidden data extraction is comparably easier than embedding. In the IWT domain, if the data is embedded in HL1 sub-band, we divide the sub-band into blocks in the same way as in embedding stage. Then we calculate the mean value of each block. If the absolute value of the mean is greater than threshold T, bit '1' is extracted. Otherwise, bit '0' is extracted. To recover the original image, we know that for bit '0' block, no manipulation is performed, while for bit '1' block, we shift back the coefficients in HL1 and LL1 sub-band. As a result the original IWT coefficients are recovered. It is noted that for hidden data extraction and original image recovering, there is no need to go back to spatial domain. It is very convenient for JPEG2000 images.

Discussion

Till now, we assume the mean value of a block is zero or close to zero, i.e., the absolute value is less than threshold T . This is true in most of cases (Refer to Figure 1). However, there do exist blocks that have mean absolute values greater than T . For these cases, we make the following discussion.

In the watermark extraction stage, bit '1's will be retrieved from these blocks since the absolute mean value is larger than threshold T . Because of the amount of those blocks is not large, we can rely on ECC to correct the possible errors. To ensure reversibility we apply different manipulation rules for different categories. Some blocks may need additional information to be recorded. The additional information is referred to as side information.

For Type A blocks:

We simply increase the absolute mean value. To recover the original image, we can shift the mean value back after data extraction. No side information is needed.

For Type B blocks:

If the mean is less than $-T$, we decrease the mean value further in data embedding. Thus no overflow will occur. To recover the original image, we can shift the mean value back after the data extraction. No side information is needed. If the mean is greater than T , we keep this block unchanged and record the block position as side information. In original image recovering stage, we know that from the side information this block is not changed. So we do not do the shift-back operation.

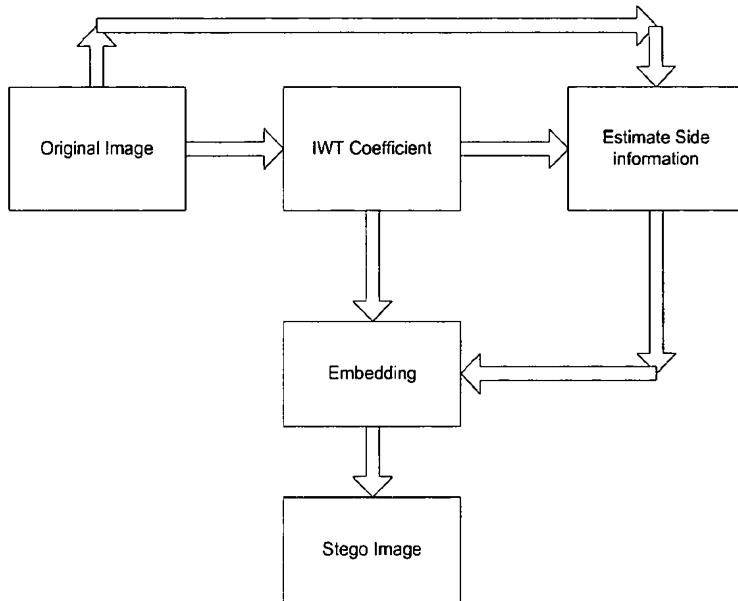
For Type C blocks:

If the mean is greater than T , we increase the mean further. No underflow will occur. To recover the original image, we can shift it back. No side information is needed. If the mean is less than $-T$, we keep this block unchanged and record the block position as side information. In the original image recovering stage, we know that from side information this block is not changed. So we do not do the shift-back operation.

For Type D block:

We keep this block unchanged and record the block position as side information. In the original image recovering stage, we know that from side information this block is not changed. So we do not do the shift-back operation.

System Framework (Embedding)



Simulation Result

1. Regular image

ECC: (15,7)
 Sub-band: HL1
 Block size: 10
 Payload: 291 bits
 Shift: 8

Table 2: Test Result for Regular Images

| | PSNR (dB) | JPEG2000 compression bit rate | Side information (number of blocks that need position recorded) |
|--------|--------------|-------------------------------------|---|
| Lena | 38.26 | 0.8 | 0 |
| Bamboo | 38.24 | 2.2 | 0 |
| Boat | 38.33 | 1.2 | 5 |

2. Eight medical images

ECC: (15,7)

Sub-band: HL1

Block size: 10

Payload: 291 bits

Shift: 8

Table 3: Test Result for Medical Images

| | PSNR | JPEG2000 Compression Bit rate | Side Information |
|---------|-------|-------------------------------|------------------|
| Min | 38.17 | 0.4 | 1 |
| Max | 38.79 | 2.4 | 15 |
| Average | 38.51 | 0.9 | 5 |

3. 81 images from CorelDraw database

ECC: (31,6)

Sub-band: HL1

Block size: 10.

Payload: 183 bits

Shift: 8

Table 4: Test Result for CorelDraw Images

| | PSNR | JPEG2000 Compression Bit rate | Side Information |
|---------|-------|-------------------------------|------------------|
| Min | 38.17 | 0.4 | 0 |
| Max | 38.79 | 2.4 | 90 |
| Average | 38.51 | 0.9 | 16 |

4. Eight JPEG2000 test images (N1A-N8A)

Various ECC coding and block size are used. Thus, payload is different for different images.

Table 5: Test Result for JPEG2000 Test Images

| | PSNR | JPEG2000 compression bit rate | Side information (Number of blocks need to be recorded) | Payload (no. of information bits) |
|-----|-------|-------------------------------|---|-----------------------------------|
| N1A | 42.67 | 0.8 | 13 | 697 |
| N2A | 41.13 | 1.4 | 39 (507 bits) | 225 |
| N3A | 40.88 | 0.8 | 8 | 697 |
| N4A | 39.50 | 0.4 | 17 | 400 |
| N5A | 39.80 | 0.6 | 50 | 400 |
| N6A | 40.77 | 0.4 | 4 | 697 |
| N7A | 43.97 | 1.6 | 29 | 400 |
| N8A | 37.16 | 1.8 | 60(780 bits) | 172 |

Side Information

For most images, there needs little side information. However, in Table 5, we can see that the side information for N2A and N8A is large. The side information could be JBIG compressed to less than 2k and 4k for these two images, respectively. According to JPEG 2000 standard draft Part 2 [5], in Annex M, Tables M2, M3, and M4, it is specified that the length for *Metadata* and *Intellectual Property Rights Information* can be variable. We argue that 2k/4K bits of side information for N2A/ N8A can be put into the header of JPEG2000 files.

Additional Testing Results

The method has been applied to frequently used test images, 1096 images in the database of commercial software CorelDRAW and eight JPEG2000 test images. The five original and marked image pairs are shown below. The test results are listed in Table 1. Clearly, there is no annoying salt-pepper noise for all of these images. It is also shown that the technique is robust against JPEG compression.



Figure 3 (a) Original Lena image.



Figure 3 (b) Marked Lena image.



Figure 4 (a) Original Baboon image.



Figure 4 (b) Marked Baboon image.



Figure 5. (a) Original Boat image.



Figure 5 (b) Marked Boat image.

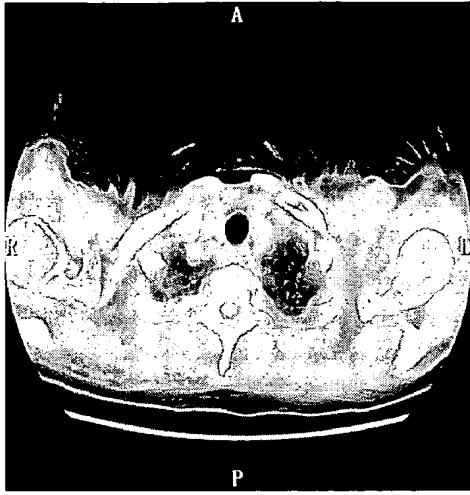


Figure 6 (a) Original medical image 1.

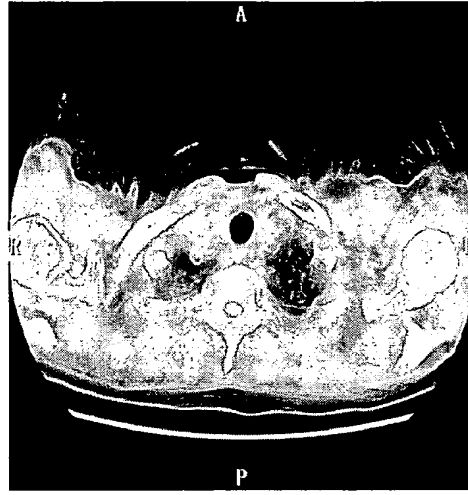


Figure 6 (b) Marked medical image 1.

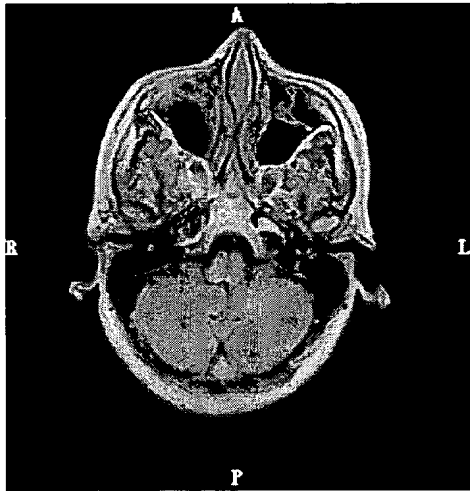


Figure 7 (a) Original medical image 2.

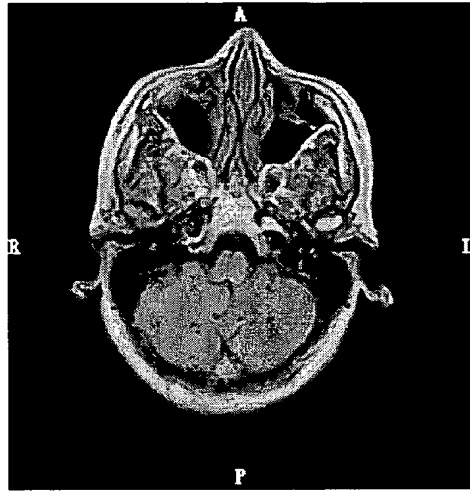


Figure 7 (b) Marked medical image 2.

| Images (512x512x8) | PSNR of marked image (dB) | Lower bound of survived data rate (bbp) |
|-----------------------|------------------------------|---|
| Lena | 38.26 | 0.8 |
| Baboon | 38.24 | 2.2 |
| Boat | 38.33 | 1.2 |
| Medical image 1 | 38.79 | 0.8 |
| Medical image 2 | 38.62 | 0.6 |

Table 1 Some experimental results (300 information bits embedded in 512x512 gray images, bpp denotes bits per pixel).

For JPEG2000 test images (N1A-N8A), test results are listed in Table 2. Figure 8 shows original image and marked image of N1A.

| Images (1536x1920x24) | PSNR of marked image (dB) | Data rate survived (bbp) | Payload (Information Bits) |
|--------------------------|---------------------------------|-----------------------------|----------------------------------|
| N1A | 42.67 | 0.8 | 697 |
| N2A | 44.28 | 1.6 | 514 |
| N3A | 40.88 | 0.8 | 697 |
| N4A | 39.50 | 0.4 | 400 |
| N5A | 39.80 | 0.6 | 400 |
| N6A | 40.77 | 0.4 | 697 |
| N7A | 43.97 | 1.6 | 400 |
| N8A | 41.25 | 1.8 | 729 |

Table 2 Experimental results for JPEG2000 test images (1536x1920, bbp denotes bits per pixel).



Figure 8 (a) Original N1A.



Figure 8 (b) Marked N1A.

REFERENCES

[fridrich 01] J. Fridrich, M. Goljan and R. Du, "Invertible authentication," *Proc. SPIE, Security and Watermarking of Multimedia Contents*, San Jose, CA, January (2001)

[goljan 01] M. Goljan, J. Fridrich, and R. Du, "Distortion-free data embedding," *Proceedings of 4th Information Hiding Workshop*, Pittsburgh, PA, April, 2001.

[honsinger 99] C. W. Honsinger, P. Jones, M. Rabbani, and J. C. Stoffel, "Lossless recovery of an original image containing embedded data," US Patent application, Docket No: 77102/E-D (1999)

[macq 99] B. Macq and F. Deweyand, "Trusted headers for medical images," *DFG VIII-D II Watermarking Workshop*, Erlangen, Germany, Oct. 1999.

[vleeschouwer 01] C. de Vleeschouwer, J. F. Delaigle and B. Macq, "Circular interpretation on histogram for reversible watermarking," *IEEE International Multimedia Signal Processing Workshop*, Cannes, France, pp.345-350, October 2001.

[xuan 02] G. Xuan, J. Chen, J. Zhu, Y. Q. Shi, Z. Ni, W. Su, "Distortionless Data Hiding Based on Integer Wavelet Transform," *IEEE International Workshop on Multimedia Signal Processing*, St. Thomas, US Virgin islands, December 2002.

[ni 03] Z. Ni, Y. Q. Shi, N. Ansari and W. Su, "Reversible data hiding," *IEEE International Symposium on Circuits and Systems*, Bangkok, Thailand, May 2003.

[1] C. Christopoulos, A. Skodras, T. Ebrahimi, "The JPEG2000 Still Image Coding System: An Overview"; *IEEE Trans. On Consumer Electronics*, Vol.46, No.4, pp.1103-1127, Nov. 2000

[2] A. Skodras, C. Christopoulos, T. Ebrahimi, "The JPEG2000 Still Image Compression Standard," *IEEE Signal Processing Magazine*, pp36-58, Sept.2001

[3] L. Gall and A. Tabatabai, "Subband Coding of Digital Images Using Symmetric Short Kernel Filters and Arithmetic Coding Techniques," *Proc. IEEE ICASSP*, NY, 1988, pp.761-765

[4] A.R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo, "Wavelet Transforms That Map Integers to Integers," *Technical report, Department of Mathematics, Princeton University*, 1996

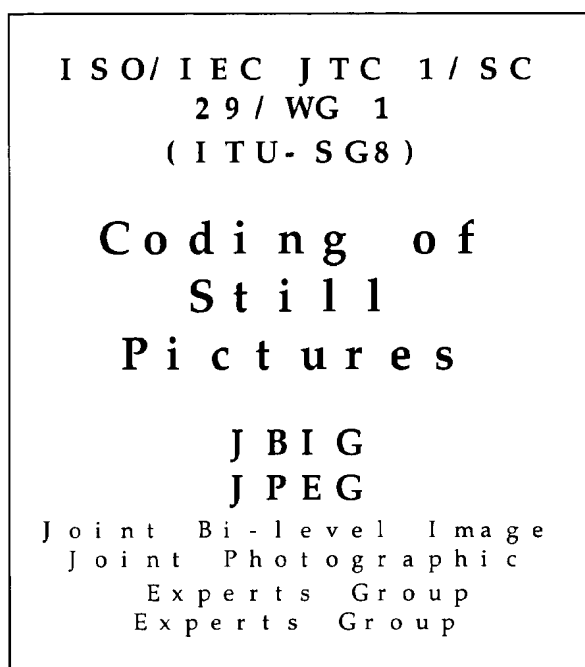
[5] JPEG2000 Standard Part 2: <http://www.jpeg.org/public/fcd15444-2.pdf>

Claims

1. Methods and apparatus as described hereinabove.

Appendix A

What follows is a technical paper that provides additional information related to our invention above. It is our understanding that as of this filing, this paper has not been published or otherwise disclosed publicly.



TITLE: A Unified Authentication Framework for JPEG2000:
Technical Description

SOURCE: Zhishou Zhang, Gang Qiu, Qibin Sun and Xiao Lin
(Institute for Infocomm Research, Singapore)
Zhicheng Ni and Yun-Qing Shi
(New Jersey Institute of Technology, USA)

PROJECT: JPEG-2000 Part-8 (JPSEC)

STATUS: Proposal

Table of Contents

| | |
|------|------------------------------------|
| 1. | Introduction..... |
| 2. | Terms and Definitions..... |
| 3. | Functionalities..... |
| 3.1. | Fragile Authentication |
| 3.2. | Lossy Authentication |
| 3.3. | Lossless Authentication |
| 4. | General Description |
| 4.1. | Fragile Authentication |
| 4.2. | Lossy Authentication |
| 4.3. | Lossless authentication |
| 5. | System Analysis..... |
| 5.1. | Complexity..... |
| 5.2. | Storage overhead..... |
| 5.3. | Data expansion..... |
| 5.4. | Impact of transmission error |
| 5.5. | Security Analysis |
| 6. | Summary |
| 7. | Reference |

Introduction

Traditional digital signature techniques (e.g., DSA or RSA) provide an effective and secure solution for data authentication, which covers both integrity protection and non-repudiation. Any one-bit modification will make the protected data unauthentic, which is definitely advantageous for data as every bit of data is vital. For example, if a transaction is made on-line, the exchanged data may contain information like amount of payment, account number and payee's name. As you can imagine, any modification, even a single-bit, will lead to total failure of the transaction.

Directly applying traditional digital signature techniques on image can also provide a good protection of image data, but in an unreasonably strict way. Such authentication on image is called fragile authentication. As images are exchanged between different entities within different media, they are unavoidably experiencing incidental distortion introduced by image transcoding, unreliable carrier and multi cycles of encoding-decoding. Although the incidental distortion changes image data, it doesn't change meaning of the image from human's point of view. An unattacked image that experienced incidental distortion will be declared as unauthentic with traditional digital signature based authentication scheme. Therefore, the fragility of traditional digital signature techniques limits the typical applications of image.

JPEG2000 has many advanced features, including lossy-to-lossless compression, better compression ratio, resolution scalability, quality scalability, ROI and so on. Therefore we should have the following principles in our mind when we design an authentication system for JPEG2000. They are:

- ◆ The authentication framework must be able to exploit advanced features of JPEG2000. For instance, the solution should be able to protect the JPEG2000 image in a scalable way. To align with JPEG2000, the solution must be able to protect any one or more components, tiles, resolutions, quality layers, ROIs, precincts, or code blocks.
- ◆ The authentication framework should be able to provide effective and secure protection for JPEG2000 images, while being robust enough for incidental distortion.
- ◆ The authentication framework should not be obtained in a way either by compromising those advanced features of JPEG2000 or by narrowing its typical applications. For instance, the solution should retain the lossless feature of JPEG2000 image.
- ◆ The authentication framework should be compatible with state-of-arts information security standards such as X.509, etc.

In the document N2946 and N3074, we have already proposed a unified content-based authentication framework for JPEG2000 images, and have given system description and test results. This document is to give more technical details of our proposed authentication system.

Terms and Definitions

| | |
|--|---|
| Authentication | Authentication is the process to protect integrity of data and to prevent repudiation. Usually It involves signing process and verification process. |
| Digital Signature | Digital signature is a natural tool for data authentication. General speaking, it should comprise some data (i.e., the signature) that the receiver can keep as evidence that a particular message was sent and that the signer was the originator. |
| Sign | Sign is the process of generating a signature for the protected data. |
| Verify | Verification is the process of detecting any possible attacks from the protected data. |
| Hash | It means one-way hash function in cryptography. Typical hash functions are MD-5 and SHA-1. |
| Lowest Authentication Bit Rate (LABR) | It refers to the authentication strength. As long as the bit-rate of the re-coded or transcoded JPEG2000 is greater than the LABR, its authenticity is guaranteed by our proposed solution. |
| Fragile Authentication | With Fragile Authentication, protection is based on image data instead of image content. Any even one-bit modification within the protected image data will make the image unauthentic, even if it doesn't change its content meaning. |
| Lossy Authentication | With Lossy Authentication, protection is based on image content. Lossy Authentication is robust against the defined incidental distortions by watermarking the content in a lossy way. |
| Lossless Authentication | With lossless authentication, protection is also based on image content, and it's also robust against the defined incidental distortions. But it can recover the original image after watermark extraction, if no transcoding is applied. |
| Incidental Distortion | Incidental distortion is introduced by common image processing and unreliable network transportation. Normally, incidental distortion doesn't change the image meaning but degrade the image quality. |
| Intentional Distortion | Intentional distortion is introduced by some kind of malicious attack, which changes the meaning of image content. |
| Lossy watermarking | Watermarking will permanently cause the degradation of image quality, though it is imperceptible. |
| Lossless watermarking | Watermarking will cause the degradation of image quality, though it is imperceptible. However, the original content can be exactly recovered after watermark extraction, if no transcoding is applied on the watermarked JPEG2000 image. |
| Error Correction Coding (ECC) | A coding system that incorporates extra parity bits in order to detect and correct errors. |
| Parity Check Bits (PCB) | The redundancy bits in order to detect and correct errors. |
| Attack | In robust authentication (lossy or lossless), it refers to any content modifications which result in a change of content meaning. In fragile authentication, any one-bit change will be deemed as attack. |

* Note that the terms and definitions list here are just for the purpose of understanding this document and may not be the same as their formal ones.

Functionalities

The proposed system integrates fragile authentication, lossy authentication and lossless authentication in one single unified framework for JPEG2000 images. Similar to JPEG2000 compression strength that is quantitatively controlled by the compression bit-rate, the authentication strength could also be quantitatively specified by a parameter called “Lowest Authentication Bit Rate (LABR)”. It means that all data/content of JPEG2000 image above LABR will be protected. Thus it will bring users much convenience.

Fragile authentication is used to protect one or more parts of codestream, or even the whole codestream from the main header to EOC marker. Since it's fragile, any one-bit modification of the protected part will make the image unauthentic. Lossy authentication is used to protect JPEG2000 image in a semi-fragile way, which is much more robust to incidental distortion. The image quality after lossy authentication degrades in an imperceptible way, due to watermark embedding. Similarly, lossless authentication also protect JPEG2000 image in semi-fragile way, but the original image can be recovered after watermark extraction, assuming no transcoding is applied. Typical functionalities of the proposed system are listed below.

Fragile Authentication

In fragile authentication mode, JPEG2000 image can be protected in various granularities, including the following:

- Protect the whole code stream.
- Protect part of the code stream pertaining to one or more tiles.
- Protect part of the code stream pertaining to one or more components.
- Protect part of the code stream pertaining to one or more resolution levels.
- Protect part of the code stream pertaining to one or more quality layers, defined by LABR.
- Protect part of the code stream pertaining to one or more precincts.
- Protect part of the code stream pertaining to one or more code blocks.
- Protect part of the code stream pertaining to one ROI.

Lossy Authentication

With Lossy authentication, the signature can survive incidental distortion such as transcoding and multi cycles of JPEG2000 encoding-decoding. However, if image content is intentionally modified, i.e. content meaning is changed, it will not be able to pass the verification process. As indicated by its name, it is lossy in the sense that image quality imperceptibly degrades after watermark embedding.

Similarly, the image can be protected in the following granularities:

- Protect the whole image content
- Protect image content of one or more quality layers, defined by LABR.
- Protect image content of one or more tiles.
- Protect image content of one or more components.
- Protect image content of one or more ROIs.
- Protect image content of one or more resolutions.
- Protect image content of one or more precincts.
- Protect image content of one or more code blocks.

In addition, with lossy authentication, it is able to allocate the attacked area, should the image be maliciously manipulated.

Lossless Authentication

Lossless authentication goes one step further. It could recover the original image after watermark extraction (if no any transcoding is applied). If transcoding is applied, the original may not be recovered. However, transcoded image can still be verified as authentic so long as the bit rate of the transcoded image is above the LABR. (It also provides the robustness against these incidental distortions). With lossless authentication, it is also able allocate the attacked area.

The image can be protected in the following granularities:

- Protect the whole image content.
- Protect image content of one or more quality layers, defined by LABR.
- Protect image content of one or more tiles.
- Protect image content of one or more components.
- Protect image content of one or more ROIs.
- Protect image content of one or more resolutions.
- Protect image content of one or more precincts.
- Protect image content of one or more code blocks.

General Description

Figure 1 give an illustration of the proposed system for JPEG2000 image authentication. The left part is the encoder and the right part is the decoder. The encoder accepts three sets of parameters, including encoding parameters (such as CBR, 5/3 filter or 9/7 filter, etc), original image to be encoded and authentication parameters (such as LABR, protected locations and authentication mode). Depending on the specified authentication mode, different authentication module will be invoked while the image is being encoded. If fragile authentication is specified, the “fragile sign” module is invoked to generate the signature, which is a straightforward solution with traditional crypto signature; If lossy authentication is specified, the “lossy sign” module is invoked to embed watermark into the image and generate signature, which is supposed to be more robust to incidental distortions; If lossless authentication is specified, the “lossless sign” module is invoked to

embed watermark into the image and generate signature, such that after signature verification, the image content can be exactly recovered if no transcoding is applied. If transcoding has been applied to the image, the JPEG2000 image can still be verified but cannot be exactly recovered. The final outputs are a JPEG2000 image (without watermark for fragile authentication and with watermark for lossy & lossless authentication) and its associated digital signature.

In the reverse direction, a decoder accepts four inputs: JPEG2000 image to be decoded, digital signature, public key and authentication parameters. Similarly, depending on the specified authentication mode, different verify module (fragile verify, lossy verify or lossless verify) will be invoked while the image is being decoded. The final outputs of the decoder are the decoded image, verification status and information about the attacked areas (in case that the image is maliciously manipulated). Note that after lossless verification, the decoded image will be exactly the same as the original image.

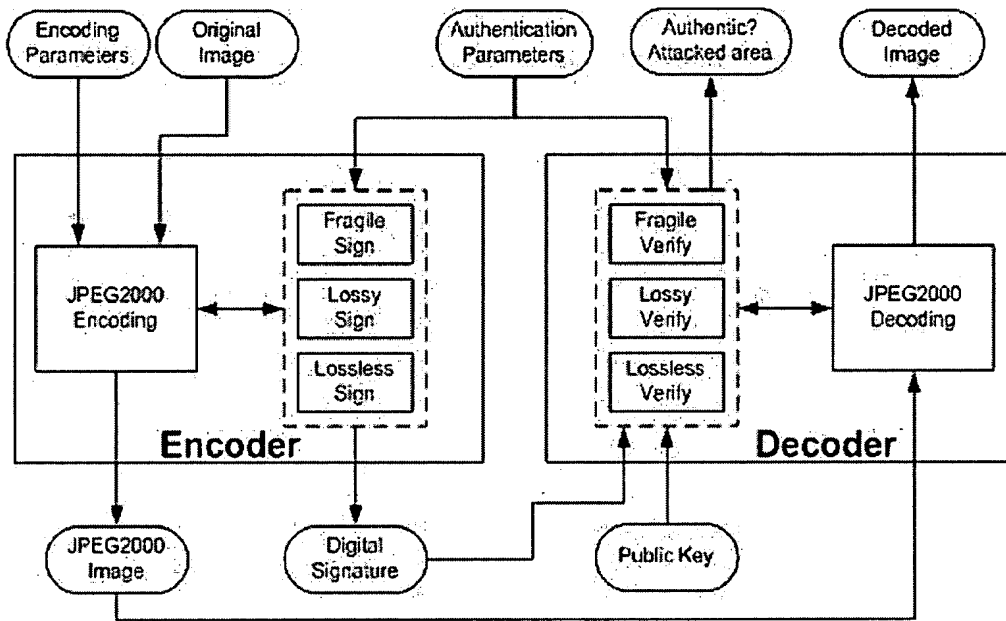


Figure 1. the diagram of proposed solution

Fragile Authentication

Fragile authentication is selected for protecting JPEG2000 image on code-streams level. Fragile signing and verifying operations are quite straightforward, as shown in Figure 2 and 3. During sign operation, the original image is encoded as per normal. While the code stream is being formulated, its protected parts, as specified by LABR and other parameters, are extracted and fed to traditional hashing and signing operation. As result, a digital signature is generated. During verify operation, while the code stream is parsed during decoding, its protected part, as specified by LABR and other parameters, is

extracted and fed to traditional hashing and verifying operation, which returns the verification result. Even one-bit change in the protected part will be deemed as unauthentic.

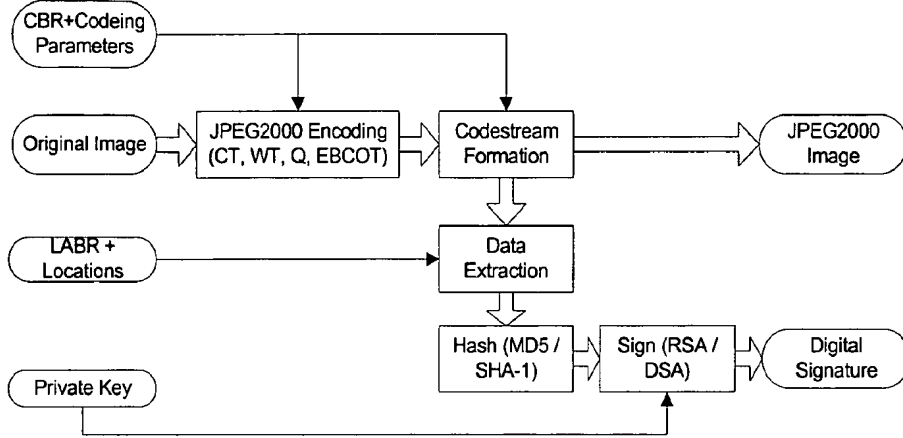


Figure 2. Fragile sign operation

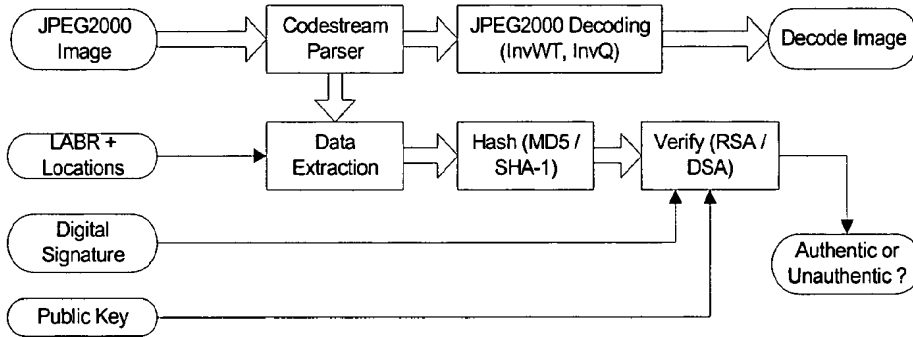


Figure 3. Fragile verify operation

Lossy Authentication

Lossy authentication is usually selected for those applications demanding for more robustness such as wireless communication. Figure 4 illustrates the basic ideas of lossy signing operation. Firstly, the original image undergoes color and wavelet transformation, quantization, arithmetic coding and EBCOT, which are all basic procedures in JPEG2000 encoding. EBCOT process will find out for each coded block those bit-planes that are above LABR (i.e., they survive transcoding operation to LABR). Then, decision is made on which resolution level (X) is suitable for feature extraction and which resolution level (Y) for watermark embedding, based on Human Vision System (HVS). The block-based feature, F_i , is then encoded with selected Error Correction Coding (ECC) Scheme to generate codeword CW_i . The Parity Check Bits of CW_i , PCB_i , is used as a seed to formulate block based watermark W_i , which is then embedded into the corresponding block in LH or HH subband of Y. In addition, features from all blocks are concatenated

and the resulted bit sequence is hashed by a cryptographic hashing function such as MD5 or SHA-1. The generated hash value can then be signed using the content sender's private key to form the crypto signature.

Figure 5 illustrates the lossy verifying operation. The codestream parser finds out for each block those bit-planes above LABR, based on which we can decide the resolution level X for feature extraction and resolution Y for watermark extraction. Block-based feature extraction is the same to that in sign operation. Block-based watermark is extracted from each block in resolution Y . Note that if the input image is not JPEG2000 format, we have to repeat the operation that is the same as the signing to obtain the watermark and the features. Then combining features and PCBs from each block forms codeword, and the whole verification decision could be made orderly. Firstly, we calculate the syndrome of the codeword for each block to see whether any blocks are uncorrectable. If yes, then we claim the image is unauthentic and those blocks with uncorrectable codewords are attacked area. However, if all codewords are correctable (i.e. errors in any feature code are correctable by its PCB), all corrected codewords are concatenated into a bit sequence, which is then cryptographically hashed. The final verification result is concluded through a cryptographic verifying operation using supplied signature and public key.

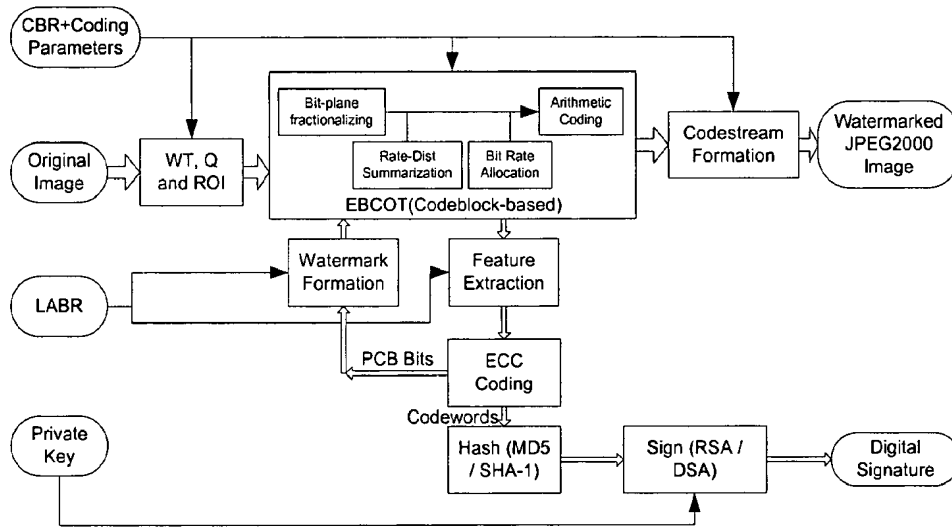


Figure 4. Lossy signing operation

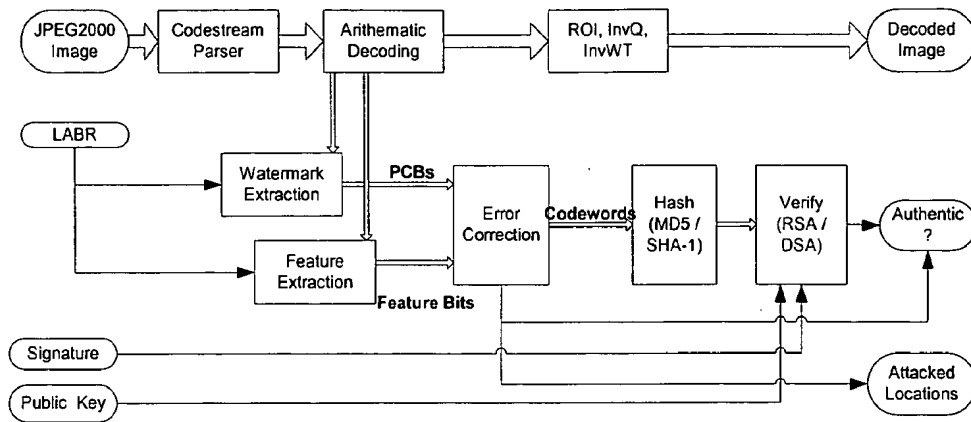


Figure 5. Lossy verifying operation for JPEG2000 file

Lossless authentication

Lossless mode is usually selected for medical or remote imaging related applications where lossless recovery of the watermarked image is required. Lossless signing operation is very similar to lossy signing operation (Figure 4). The only difference lies in watermark embedding module. The codeblock whose size is usually 64x64 is further divided into 8x8 blocks called patches. The coefficients in a patch are split into two subsets. Then we calculate the difference value α , which is defined as the arithmetic average of differences of coefficients in two respective subsets. Since in a patch, the coefficients are highly correlated, the difference value α is expected to be very close to zero. Furthermore, it has certain robustness against incidental distortions because α is based on all coefficients in the patch. Each patch is embedded with one bit, as illustrated in Figure 6. If 1 is to be embedded, we shift difference value α to right side or left side beyond a threshold, by adding or subtracting a fixed number from each coefficients within one subset. If 0 is to be embedded, the patch is intact. There are chances that the value α is originally beyond the threshold and a bit of 0 is to be embedded. In this case, we shift the value α further away beyond the threshold, and rely on ECC to correct the bit error, because the watermark bits are ECC encoded again before being embedded.

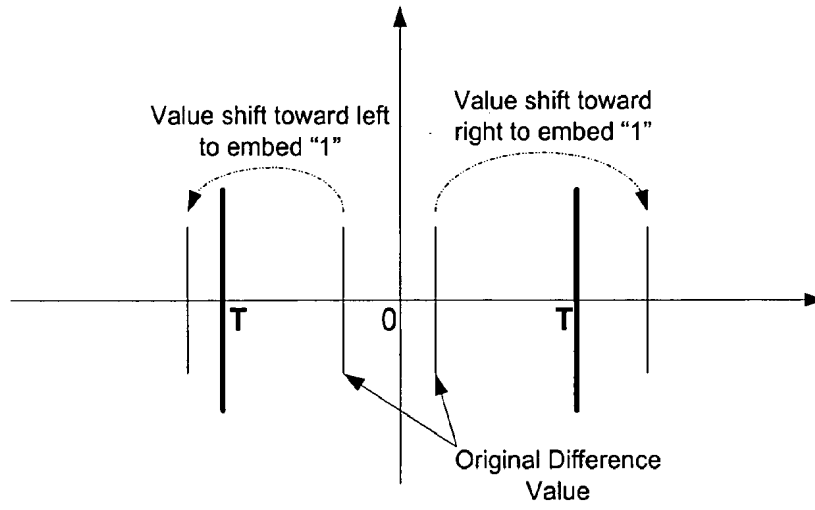


Figure 6. Embedding a bit "1"

Lossless verifying operation is also similar to lossy one, with the exception of watermark extraction. The code block is divided in patches and difference value α of each patch is calculated in the same way as lossless sign. For each patch, if value α is beyond the threshold, a bit of "1" is extracted and the difference value is shifted back to its original position, which means that original coefficients are recovered. If the value α is inside the threshold, a bit of "0" is extracted and nothing needs to be done. Finally an ECC correction is applied on the extracted bit sequence to get the correct watermark bits.

System Analysis

This section elaborates more on algorithmic complexity, storage overhead, data expansion, impact of transmission error and security analysis.

Complexity

For fragile sign and verify operation, the processing overhead is around 5% of standard JPEG2000 encoding/decoding time. It mainly comes from:

- Find from the code stream the protected segment that is specified by LABR and Location parameters, and extract it from the code stream. This can be done while the code stream is being formulated, thus the processing overhead is quite minimal.
- Perform one hash (MD5/SHA-1) operation on the extracted part of the code stream.
- Perform one RSA/DSA sign or verify operation.

For lossy and lossless operation, the process overhead is around 20% of standard JPEG2000 encoding/decoding processing time. It mainly comes from:

- Find out all bit-planes above LABR, in order to decide where to extract feature and where to embed / extract watermark.
- Extract features from each protected code block.
- ECC coding or correction for each protected code block.
- Watermark embedding or extraction
- One hash operation.
- One RSA/DSA sign or verify operation

Storage overhead

For fragile sign and verify, the only overhead comes from the fact that the protected part of code stream need to be temporarily stored in memory in order to sign/verify. For example, for 500KB image, the maximum memory overhead is 500KB.

For lossy and lossless authentication, the quantized coefficients in current tile need to be temporarily stored in memory in order not to repeat wavelet and quantization steps. In this case, the overhead depends on size of a tile. E.g., if tile size is 256x256, the overhead will be around 1 MB.

Data expansion

For fragile authentication, the overhead is only side information, like signature, protected location, public key and so on. It is roughly about 300 bytes. The resulted code stream is exactly same as normal encoding.

For lossy and lossless authentication, the overhead of side information is the same as fragile authentication. The resulted code stream size is about 0~200 bytes more or less than normal encoded code stream.

Impact of transmission error

For fragile authentication, any transmission error will resulted in failure of verification, due to the nature of traditional crypto signature.

For lossy and lossless authentication, as long as the number of error bits is not significant, our solution can still authenticate the image, due to the robustness of our solution.

Security Analysis

For Fragile authentication, the security strength is the same as that of the underlying Hash (MD5 or SHA-1) and Sign (RSA or DSA) algorithm.

However, for lossy and lossless authentication, content-based feature extraction and error correction coding (ECC) reduce the security strength, as some modification may not affect the extracted features or modified features can be corrected by ECC. However, such security risk can be compensated from image contextual characteristics.

Summary

We proposed a systematic and quantitative way for authenticating multimedia content by casting the content into a finer representation in terms of authentication bit rate. This then brings much convenience for the authentication applications by simply keying in one parameter—authentication bit-rate to protect the content.

We also proposed a framework for meeting different authentication requirements from real applications by employing different signing modules (fragile, lossless and lossy) which is in line with different JPEG2000 coding settings. The proposed scheme is fully compatible with JPEG2000 coding and traditional crypto schemes.

We believe that the proposed scheme is well suited to and needed by JPSEC tools.

Reference

- ◆ Zhishou Zhang, Gang Qiu, Qibin Sun, Xiao Lin, Zhicheng Ni, Yun-Qing Shi, WG1N3074 “A Unified Authentication Framework for JPEG2000 images: System Description and Experiment Results”
- ◆ Qibin Sun, Xiao Lin and Yun-Qing Shi, WG1N2946 “A Unified Authentication Framework for JPEG2000 images”
- ◆ Touradj Ebrahimi and Claude Rollin, , WG1N30555 “JPSEC Working Draft – Version 2.0”